

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

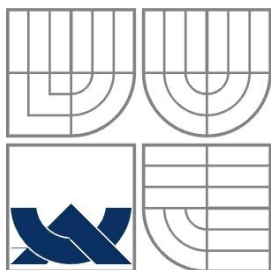
NÁSTROJ PRO MONITOROVÁNÍ A OZNAMOVÁNÍ MULTICASTOVÝCH MULTIMEDIÁLNÍCH STREAMŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

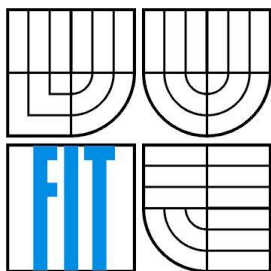
AUTOR PRÁCE
AUTHOR

PETR DORUŠÁK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

NÁSTROJ PRO MONITOROVÁNÍ A OZNAMOVÁNÍ MULTICASTOVÝCH MULTIMEDIÁLNÍCH STREAMŮ

A TOOL FOR MONITORING AND ANNOUNCING MULTICAST MULTIMEDIA STREAMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR DORUŠÁK

VEDOUCÍ PRÁCE
SUPERVISOR

MGR. JANA SKOKANOVÁ

BRNO 2013

Abstrakt

Tato práce se zabývá návrhem a implementací nástroje pro monitorování a oznamování multicastových multimediálních streamů. Cílem práce je vytvořit aplikaci, která bude monitorovat MPEG-TS streamy v zadaných multicastových skupinách a bude informovat o jejich běhu prostřednictvím oznamovacího protokolu SAP.

Abstract

This thesis deals with design and implementation of tool for monitoring and announcing of multicast multimedia streams. The aim is to create an application that will monitor the MPEG-TS streams in the specified multicast groups and will inform about their run through the Session Announcement Protocol.

Klíčová slova

Streamování, multicast, oznamování, monitorování, multimédia, SAP, SDP, MPEG-TS

Keywords

Streaming, multicast, announcing, monitoring, multimedia, SAP, SDP, MPEG-TS

Citace

Dorušák Petr: Nástroj pro monitorování a oznamování multicastových multimediálních streamů, bakalářská práce, Brno, FIT VUT v Brně, 2013

Nástroj pro monitorování a oznamování multicastových multimediálních streamů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Mgr. Jany Skokanové
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Dorušák
25.4.2013

Poděkování

Mé poděkování patří Mgr. Janě Skokanové za odbornou pomoc, poskytnuté rady a připomínky při psaní této práce.

© Petr Dorušák, 2013

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Metody komunikace v počítačových sítích.....	3
2.1 Unicast	3
2.2 Broadcast	4
2.3 Multicast	5
3 Oznamování multicastových relací.....	6
3.1 Session Announcement Protocol (SAP)	6
3.2 Session Description Protocol (SDP).....	9
4 MPEG Transport Stream (MPEG-TS).....	17
5 Návrh programu	19
5.1 Moduly.....	20
5.2 Sériovost a paralelismus	21
5.3 Konfigurace	21
5.4 Princip programu	23
6 Implementace programu	25
6.1 Zachytávání paketů.....	25
6.2 Generování a zasílání SAP paketů.....	26
6.3 Nahrazování zdroje streamů	27
7 Testování.....	29
7.1 Testování modulů	29
7.2 Zátěžové testování	30
8 Závěr	32
Příloha A.....	35
Příloha B.....	40

1 Úvod

Streaming multimédií označuje přenos nepřetržitého toku multimediálních dat po počítačové síti v reálném čase rychlostí přehrávání. Jedná se o mechanismus vhodný pro jednorázové doručení multimediálního obsahu, kdy přijímači stačí malá vyrovnávací paměť. Streaming multimédií může mít podobu tzv. živého přenosu (*Live streaming*), anebo streamingu na vyžádání (*On-demand streaming*). Při živém přenosu je stejný multimediální obsah přenášen všem příjemcům v reálném čase a přehrávání tohoto obsahu nelze nijak řídit. Naproti tomu při streamingu na vyžádání je klientovy streamem zasílán předem vytvořený multimediální obsah, jehož přehrávání je možno řídit. Nejčastějším příkladem streamingu na vyžádání je video na vyžádání (*Video on demand*).

Cílem této práce je navrhnout a implementovat aplikaci, která bude monitorovat živě přenášené streamy, které jsou vysílány na předem známých multicastových adresách a bude o jejich běhu informovat prostřednictvím oznamovacího protokolu SAP (*Session Announcement Protocol*).

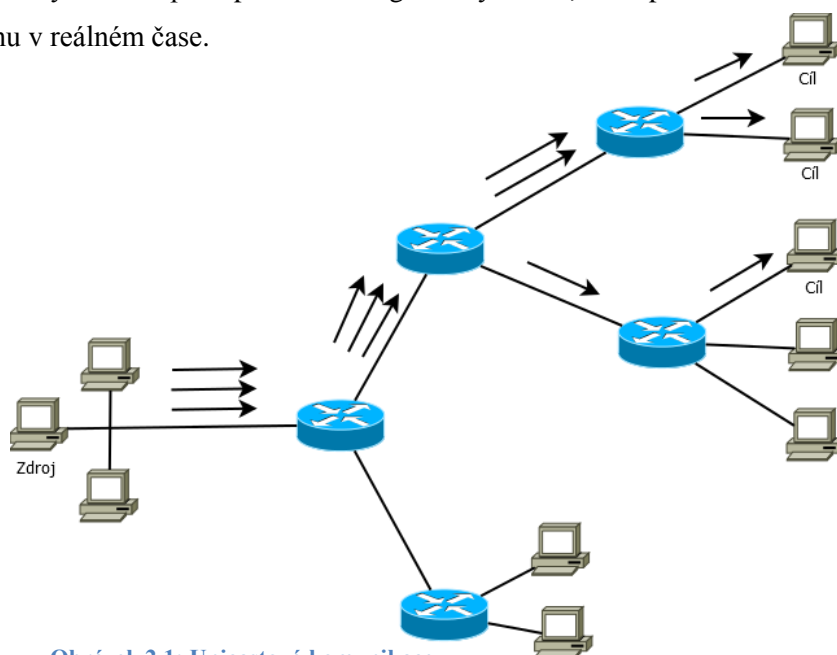
V následující kapitole se věnuji základním způsobům komunikace, které jsou využívány pro komunikaci v počítačových sítích a internetu. Dále v kapitolách 3.1 a 3.2 popisují protokoly určené pro oznamování multicastových relací, které jsou v aplikaci použity. V jádru práce popisují návrh a způsob implementace vytvořené aplikace a v závěru hodnotím možnosti jejího praktického využití.

2 Metody komunikace v počítačových sítích

V počítačové síti existují tři základní metody, které jsou využívány síťovými prvky pro jejich komunikaci. V závislosti na použité metodě se mění způsob doručování dat mezi nimi. Nejběžněji používanou metodou je tzv. unicast, umožňující doručit data jednomu konkrétnímu příjemci. Další dvě běžně používané metody jsou multicast a broadcast.¹ V této kapitole jsou všechny tři metody popsány a uvedeny jejich klady a zápory v případě použití pro streamování multimédií.

2.1 Unicast

Metoda unicast se typicky využívá pro dvoubodovou komunikaci a představuje přímé propojení zdrojového prvku s prvkem cílovým. Pokud je metoda využita pro komunikaci více než dvou bodů, je zdrojovým uzlem zaslána individuální kopie dat každému cílovému uzlu, se kterým komunikuje. Tuto situaci naznačuje obrázek 2.1. Počet cílových uzlů, se kterými může zdroj komunikovat, je v případě této komunikace závislý na šířce pásma zdrojové linky. Uvažujme o příkladu, že bychom měli 10 000 zákazníků a chtěli bychom jim umožnit přijímat televizní vysílání pomocí streamingu. Toto televizní vysílání by bylo vysíláno přenosovou rychlostí 3 Mbit/s. V tomto případě s použitím metody unicast, bychom museli mít u zdrojového uzlu připojení s datovou propustností 30 Gbit/s. To je důvod, proč je unicast využíván spíše pro streaming na vyžádání, než pro masovou distribuci multimediálního obsahu v reálném čase.



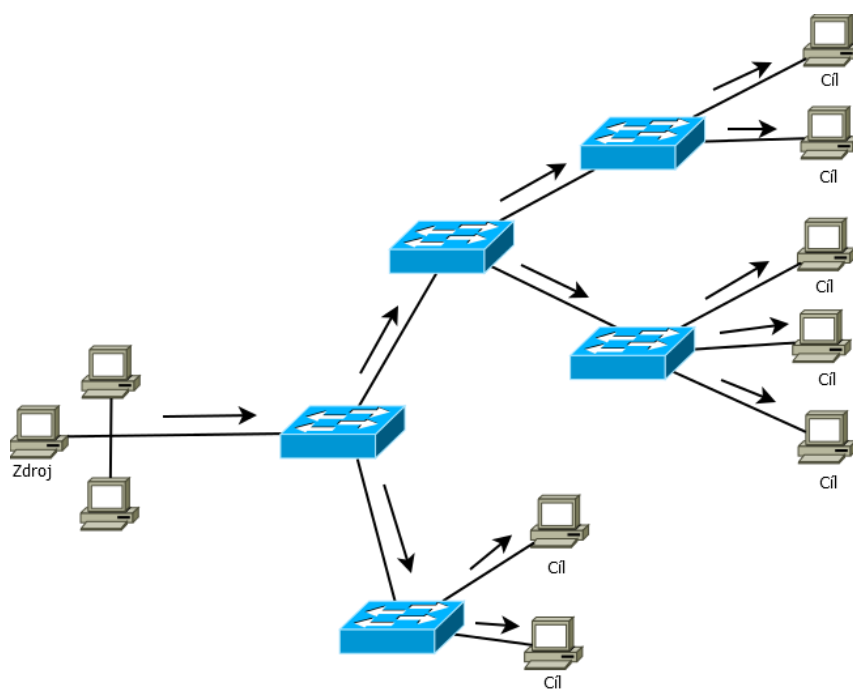
Obrázek 2.1: Unicastová komunikace

¹HARTE, Lawrence. Introduction to data multicasting. Fuquay-Varina: Althos Publishing, c2008, s. 3-7. ISBN 19-328-1355-1.

2.2 Broadcast

V případě broadcastu jde zcela o odlišný přístup, než je tomu v případě unicastu. Jedna kopie dat, kterou zdroj vysílá, je zasílána všem prvkům v dané broadcastové doméně a to bez ohledu na to, zda daný prvek o data požádal či nikoliv. Tato situace je zobrazena na obrázku 2.2.

Broadcast je v multimédiích typicky asociovaný s radiovým nebo televizním přenosem po radiových vlnách, kde je radiový signál zasílán mnoha příjemcům v určité geografické oblasti. V případě počítačových sítí je však zasílání dat všem příjemcům spíše nepříznivým vlivem. Tato nadbytečná komunikace je prvky sítě zahazována a způsobuje jejich vyšší zatížení. V extrémních případech, kdy se takto zasílá velké množství dat, může dojít k tzv. *broadcastové bouři*, tedy zahlcení sítě broadcastovými zprávami, kdy jsou síťové prvky natolik zaneprázdněné obsluhováním broadcastové komunikace, že nedokážou zpracovávat ostatní komunikaci. Proto tato metoda není příliš vhodná pro posílání multimediálních streamů po počítačové síti.

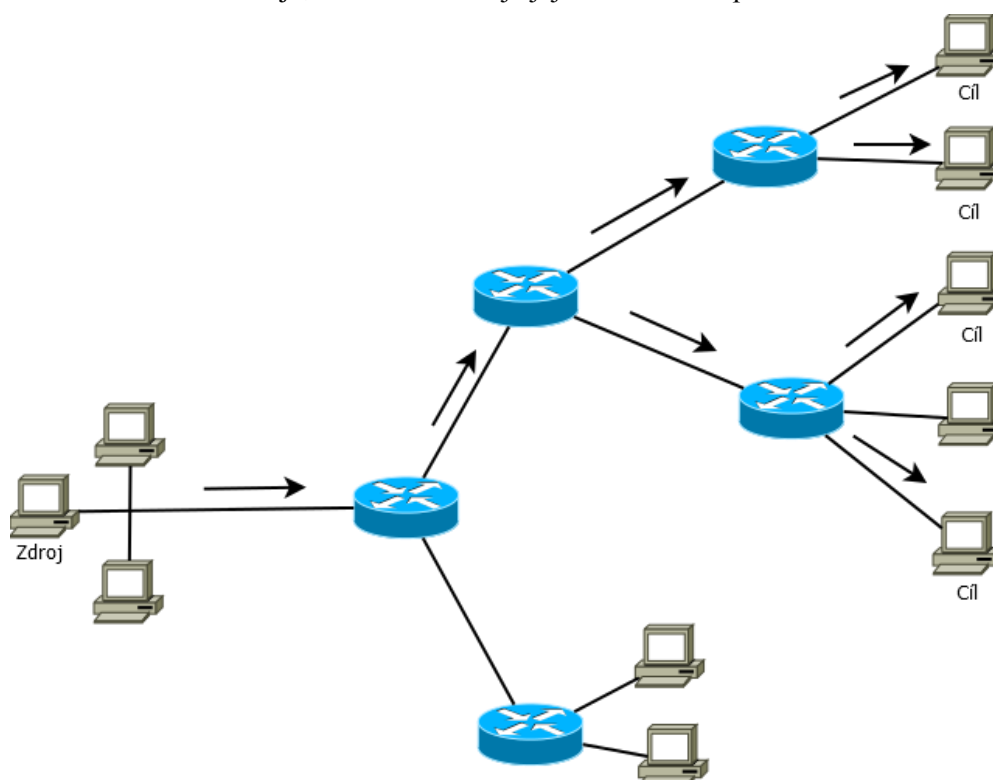


Obrázek 2.2: Broadcastová komunikace

2.3 Multicast

Multicast je obecně metoda, kdy jsou data z několika zdrojů přeposílána skupině příjemců. Využívanější variantou této metody je však ta, kdy jeden zdroj zasílá data vybrané skupině příjemců, které se říká multicastová skupina. Tato metoda oproti broadcastu snižuje vytížení sítě tím, že neposílá data těm prvkům sítě, které nemají o daná data zájem. Zájemce se pro příjem dat přihlásí do multicastové skupiny, čímž k nim získá přístup. V multicastové skupině se vyskytuje vždy pouze jediná kopie dat, která je zdrojem zasílána na adresu multicastové skupiny. Data jsou ve skupině distribuována až ke směrovači, který je nejblíže cílovému prvku, jenž o ně zažádal. Tento směrovač následně zajistí zaslání dat zájemci. Princip multicastu je zobrazen na obrázku 2.3.

Jelikož se data u této metody zbytečně nedistribuuje do částí sítě, kde o ně není zájem a tím se šetří kapacita sítě, je tato metoda ze všech zmíněných nejvhodnější pro streaming multimédií, videokonference, IPTV či internetová rádia. Problémem této metody však je, že mnohdy není podporována ze strany poskytovatelů internetového připojení. Většina poskytovatelů mezi sebou multicastová data nedistribuuje, a tak neumožňuje jejich šíření dále po síti.



Obrázek 2.3: Multicastová komunikace

3 Oznamování multicastových relací

Multicastová relace je definována jako množina multimediálních odesílatelů, příjemců a datových streamů tekoucích od odesílatelů k příjemcům. Příkladem takového multimediální relace je multimediální konference.

Existuje řada způsobů, jak získat informace o aktivních multicastových relacích. Lze je najít na internetu, mohou být zaslány prostřednictvím emailu, nebo může být účastník pozván k multicastové relaci pomocí protokolu SIP.² Informace o relaci je však možné také zjistit za pomoci oznamování. U této metody jsou potenciálním účastníkům zasílána oznámení o běžících relacích s informacemi o tom, jak se k nim připojit. Tento princip využívá dva protokoly - *Session Announcement Protocol*³ a *Session Description Protocol*⁴, které jsou v této kapitole popsány a které spolu velmi úzce souvisí.

3.1 Session Announcement Protocol (SAP)

Session Announcement Protocol je oznamovacím protokolem zasílajícím informace potenciálním účastníkům o konaných multicastových multimediálních relacích a o tom, jak se k nim připojit. Tyto informace jsou protokolem zasílány periodicky na předem známou multicastovou adresu a port (obvykle 224.2.127.254, UDP port 9875). Aby byly tyto informace vhodně strukturovány, využívá protokol SAP pro jejich prezentaci Session Description Protocol, který nese za svojí hlavičkou a který je popsán v kapitole 3.2.

3.1.1 Rozsah oznámení

Oznámení protokolu SAP jsou obvykle zasílána do stejného rozsahu, jako mají relace, o kterých informují. To zajišťuje, že příjemci oznámení se stanou pouze ti účastníci, kteří se mohou k dané multicastové relaci opravdu připojit. Tento rozsah je většinou definován pomocí TTL (*Time-To-Live*), které je obsaženo v hlavičce IP protokolu oznamovacího paketu.

SAP je navržen za účelem oznamování existujících, dlouhotrvajících, široko oblastních relací. Relace jsou oznamovány periodickým zasíláním multicastu, jehož doba opakování může být až v řádech desítek minut. To vede k tomu, že aplikačním programům zpracovávajícím protokol SAP, bude trvat jistou dobu, než zobrazí všechna oznámení o probíhajících relacích. Pro snížení této doby

² ROSENBERK, J., SCHULYRINE, H., CAMMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, R., SCHOOLER, E. *SIP: Session Initiation Protocol*, RFC3261, June 2002.

³ HANDLEY, M., PERKINS, C., WHELAN, E. *Session Announcement Protocol*, RFC2974, October 2000.

⁴ HANDLEY, M., JACOBSON, V., PERKINS, C. *SDP: Session Description Protocol*, RFC4566, July 2006.

je doporučeno nasazení proxy cache, která identifikuje veškerá SAP oznámení v rozsahu a potvrzuje aktuálnost seznamu relací.

3.1.2 Formát paketu

Oznamovací paket protokolu SAP obsahuje vícepoložkovou hlavičku, za kterou následuje v datové části SDP protokol. Aby mohla být ověřena pravost neseného SDP, může paket nést také autentizační hlavičku. Formát hlavičky je zobrazen v tabulce 3.1.2.

0		3	4	5	6	7	8			15	16		31
V=1		A	R	T	E	C	Auth len				Message id hash		
Originating Source (32 nebo 128 bitů)													
Optional authentication data													
Optional payload type													
										0			
Payload													

Tabulka 3.1.2: Formát SAP paketu

Hlavička začíná vždy dvoubitovým polem, jež označuje verzi použitého protokolu. Tato položka má vždy hodnotu 1 a to i v případě, že užíváme druhou verzi protokolu. Je to dáno tím, že druhá verze protokolu je zpětně kompatibilní s první verzí. Za verzí protokolu následuje skupina pěti příznakových bitů:

- *Address Bit (A)* – určující, zda má adresa originálního zdroje (*Originating Source*) velikosti 32 bitů (IPv4), nebo 128 bitů (IPv6),
- *Reserved (R)* – který je rezervovaný a má hodnotu 0,
- *Message Type (T)* – označující typ vysílaného paketu, tedy, zda se jedná o oznamovací paket, či paket rušící relaci,
- *Compressed Bit (C)* – definující, zda jsou data protokolu SDP komprimovaná,
- *Encryption Bit (E)* – který říká, zda jsou data protokolu SDP zašifrovaná.

Za touto pěticí následuje položka *Authentication Length*, která definuje velikost autentizační hlavičky, pokud ji paket obsahuje. Tato hlavička se v případě nenulové hodnoty *Authentication Length* nachází v oblasti *Optional Authentication Data*. Následující položkou je *Message Identifier Hash*, což je 16 bitový unikátní identifikátor oznámení. Poslední povinnou položkou hlavičky je

Originating Source neboli IP adresa originálního zdroje zprávy. Tato adresa má tvar IPv4 adresy, pokud je *Address bit* nulový, jinak je v tomto poli nachází IPv6 adresa. Za ní následují dvě volitelné části a to již zmíněná *Optional Authentication Data*, která obsahuje autentizační hlavičku a *Optional Payload Type* popisující formát zprávy nesené v části *Payload*.

Ve standardu RFC 2974⁵ je doporučeno, aby velikost SAP paketu nepřesahovala velikost větší než 1 kB, a to proto, aby nedocházelo k fragmentaci paketu. Fragmentace má totiž ztrátový efekt, který výrazně ovlivňuje spolehlivost doručení oznámení.

3.1.3 Změna a zrušení oznámení

Protokol SAP umožňuje modifikaci či odstranění již vytvořeného popisu relace. Relace, která již byla oznámena, může být modifikována jednoduše a to oznámením relace s modifikovaným SDP. U tohoto nového oznámení však musí být změněn *Message Id Hash*, aby příjemci věděli, že došlo ke změně zprávy. Dále je doporučeno, aby upravené oznámení obsahovalo autentizační hlavičku, která bude podepsaná stejným klíčem jako originální oznámení, a to z důvodu předejití neautorizované modifikaci. Alternativou k tomuto postupu je, že obě oznámení neobsahují autentizační hlavičku, ale pocházejí od stejného zdroje.

V případě zrušení relace se postupuje podobným způsobem. Je zaslán tzv. *Session Deletion Paket* se stejnou verifikační procedurou, která je použita pro modifikaci relace. Oznámení může být ale také smazáno za pomoci explicitního či implicitního timeoutu. V prvním případě obsahuje SDP uvnitř SAP protokolu časovou informaci o začátku a konci relace, po jejíž překročení může dojít k odstranění oznámení. Ve druhém případě není konec relace obsažen v SDP paketu a očekává se, že oznamovací zprávy budou zasílány periodicky. Pokud není oznámení znovu zasláno během jistého časového okamžiku, obvykle deseti časových period nebo během jedné hodiny, je oznámení relace odstraněno. Z těchto dvou časových intervalů je vybrán ten, jehož doba je větší.

⁵ HANDLEY, M., PERKINS, C., WHELAN, E. *Session Announcement Protocol*, RFC2974, October 2000.

3.2 Session Description Protocol (SDP)

Session Description Protocol je protokol navržený pro popis multimediálních relací za účelem jejich oznamování, pozvání k relacím a navázání jiných forem multimediálních relací. Jedná se o textový protokol, který slouží čistě k popisu relace a poskytuje informace o tom, jak se k dané relaci připojit. Není vázán na konkrétní transportní či aplikační protokol, je naopak nesen jinými protokoly.

3.2.1 Přenos SDP

Data SDP bývají uložena v datové části různých protokolů, jako jsou například *SAP*, *SIP*, *RTSP*, popřípadě mohou být nesená, za pomoci *e-mailu s rozšířením MIME* a *HTTP* protokolu. Pokud je SDP nesen pomocí e-mailu či HTTP protokolu, může se oznámení, které nese, dostat většímu počtu příjemců, než jak je tomu v případě zasílání SAP protokolem. To je dáno tím, že SAP posílá oznámení pouze legitimním účastníkům, kteří se nacházejí v dosahu TTL oznamovacího paketu či v adresovém rozsahu oznamovací zprávy.⁶

3.2.2 Obsah SDP

SDP obsahuje *název a účel relace*, *čas*, po který je relace aktivní, *média*, která relace obsahuje a *informace potřebné k příjmu* těchto médií (adresy, porty, formáty, atd.). Dalšími informacemi, které mohou být také příjemci žádané, jsou *informace o velikosti datového toku* použité při streamingu konferencí a *kontaktní informace* na osobu zodpovědnou za relaci. SDP musí zprostředkovat tyto informace, aby umožnilo aplikacím připojit se k vysílané relaci.

Položky SDP se dají rozdělit do dvou základních částí. *Session-level description* a *Media-level description*. *Session-level description* obsahuje položky, které jsou použity na celou relaci a všechna média popsaná v tomto SDP. *Media-level description* je volitelná část, která se může vyskytovat v SDP několikrát a detailně vždy popisuje jedno médium.

⁶ KOSIUR, David. *IP multicasting*. Vyd. 1. New York: J. Wiley, 357 s. ISBN 0-471-24359-0.

3.2.3 Formát a význam parametrů

Všechny parametry v SDP protokolu jsou uvedeny ve formátu: *<TYP>=<HODNOTA>*, kde *TYP* je právě jedno písmeno a *HODNOTA* je strukturovaný řetězec, jehož formát závisí na položce *TYP*. V následujících podkapitolách je uveden přesný význam a struktura jednotlivých parametrů SDP protokolu.

3.2.3.1 Protocol Version (v=)

Formát: *v=0*

Parametr Protocol Version je povinný, určuje verzi SDP protokolu a nabývá vždy hodnoty 0.

3.2.3.2 Origin (o=)

Formát: *o=<username> <session id> <session version> <netw. type> <address type> <address>*

Origin je povinný parametr, který určuje původce neboli zdroj relace. Ten je dán jeho uživatelským jménem (*username*) a IP adresou (*address*), dále identifikátorem relace (*session id*) a číslem verze relace (*session version*). Kombinace těchto částí tvoří globálně jednoznačný identifikátor relace, pomocí kterého je relace identifikována. *Session version* je číslo verze, které je zvětšováno při každé modifikaci informací o relaci. Část *network type* definuje typ počítačové sítě, v níž je SDP vysílán (typicky „IN“ - Internet). *Address type* definuje typ IP adresy v položce *address* a může nabývat hodnot „IP4“ a „IP6“, pokud se jedná o IPv4 respektive IPv6 adresu.

Příklad: *o=- 2072231115113 26 IN IP4 147.229.10.86*

3.2.3.3 Session Name (s=)

Formát: *s=<session name>*

Session Name je textová položka, která musí být v popisu relace právě jednou a určuje název relace.

Příklad: *s= Vysílání z D0206 FIT VUT v Brně*

3.2.3.4 Session and Media Information (i=)

Formát: *i=<session description>*

Tento parametr slouží k podání textových informací o relaci či médiu. Položka se může vyskytovat maximálně jednou v *Session-level description* sekci a jednou v každé *Media-level description* sekci.

Příklad: *i= Vysílání přednášky ISA z učebny D0206 FIT VUT v Brně*

3.2.3.5 URI (u=)

Formát: *u=<URI>*

URI je Universal Resource Identifier⁷, který slouží k odkázání na další informace o relaci. Tato položka je volitelná a smí se vyskytovat v popisu relace pouze jednou na pozici před specifikací prvního média.

Příklad: *u=www.dalsiinfo.cz*

3.2.3.6 Email Address (e=) a Phone Number (p=)

Formát: *e=<email address>* *p=<phone number>*

Tyto volitelné parametry slouží k podání kontaktních informací o osobě zodpovědné za vysílanou relaci. Jedná se o e-mailovou adresu a telefonní číslo na danou osobu. Telefonní číslo zadané v této položce by mělo obsahovat mezinárodní předvolbu.

Příklad: *e=admin@stream.com* *p=+420 605 111 222*

⁷ BERNERS-LEE, T., FIELDING, R., MASINTER, L. *Uniform Resource Identifier (URI): Generic Syntax*, RFC3986, January 2005

3.2.3.7 Connection Data (c=)

Formát: *c=<network type> <address type> <connection address>*

Položka Connection Data poskytuje informace potřebné pro připojení účastníka do relace. Jsou jimi IP adresa nebo seznam IP adres, ke kterým se klient připojuje pro sledování relace, typ této IP adresy/adres a typ sítě, ve které se relace nachází. Connection Data patří k jedné z nejdůležitějších položek protokolu SDP. Jedná se o první část potřebnou pro sledování relace. Druhou částí je číslo portu transportního protokolu, které se nachází v parametru *Media Announcement*. Proto se parametr Connection Data musí nacházet jednou v sekci *Session-level description*, nebo musí být uváděn pro každé médium zvlášť.

Network type definuje typ počítačové sítě, v níž je SDP vysílán. Typicky má hodnotu „IN“ jako Internet. *Address type* informuje o typu IP adresy v položce *connection-address*. Jedná-li se o IPv4 adresu, má hodnotu „IP4“, v případě IPv6 adresy, bude hodnota „IP6“. IP adresa či seznam IP adres se skrývají pod částí *connection-address*. Pokud se jedná o běžnou unicastovou či multicastovou IPv4 adresu, je za ní uvedeno TTL, které je od ní odděleno lomítkem (např: *c=IN IP4 224.2.1.1/127*). To je zde přidáno za účelem určení rozsahu, v němž se budou pakety relace zasílat. V případě IPv6 adresy TTL uvedeno není, protože se očekává, že rozsah relace bude omezen za pomoci adresových rozsahů IPv6. V případě seznamu adres lze seznam zapsat dvěma způsoby. Můžeme celou položku zapsat několikrát s různými adresami jako seznam, nebo jako jedinou položku za pomoci lomítka a počtu adres následujících za TTL, nebo za adresou v případě IPv6. Zde jsou uvedeny dva příklady parametru, které jsou identické:

Př1: *c=IN IP4 224.2.1.1/127*
c=IN IP4 224.2.1.2/127
c=IN IP4 224.2.1.3/127

Př2: *c=IN IP4 224.2.1.1/127/3*

c=IN IP6 FF15::101
c=IN IP6 FF15::102
c=IN IP6 FF15::103

c=IN IP6 FF15::101/3

3.2.3.8 Bandwidth (b=)

Formát: *b=<bwtype>:<bandwidth>*

Bandwidth je volitelným parametrem, který specifikuje vhodnou šířku pásma pro přenos relace nebo určitého média. *Bandwidth* je hodnota šířky pásma uvedená v kbit/s. *Bwtype* je alfanumerický

identifikátor, který dává význam hodnotě *bandwidth*. V současné době jsou specifikovány dva identifikátory:

- CT Conference Total, který specifikuje navrhovanou horní mez šířky pásma pro celou relaci,
- AS Application-Specific, který specifikuje navrhované horní meze šířky pásma pro jednotlivá média.

Příklad: *b=AS:5610*

3.2.3.9 Timing (t=)

Formát: *t=<start-time> <stop-time>*

Parametr Timing je povinný parametr, který popisuje začátek a konec trvání relace. Pokud je relace aktivní vícekrát v nepravidelně opakujících se časech, pak by měl být tento parametr použit v popisu relace několikrát. Pokud relace probíhá v pravidelně opakujících se časech, je doporučeno použít parametr *Repeat Times* spolu s tímto parametrem. *Start-time* popisuje začátek běhu relace a *stop-time* konec běhu relace. Obě tyto části jsou desetinné zastoupení NTP protokolu⁸. Jedná se o 64 bitové hodnoty, které popisují čas v sekundách od roku 1990. Je-li hodnota *stop-time* nulová, není relace aktivní, dokud nenastane čas *<start-time>*. Pokud je i hodnota *start-time* nulová, je relace označena za trvale běžící.

Příklad: *t=0 0*

3.2.3.10 Repeat Time (r=)

Formát: *r=<repeat interval> <active duration> <offsets from start-time>*

Repeat Time je volitelný parametr, který se jako parametr *Timing* může vyskytovat v popisu relace několikrát a popisuje časy opakování relace. Jednotkami všech položek tohoto parametru jsou sekundy. Pokud například budeme mít relaci, která bude běžet každé pondělí v 10 hodin ráno a v úterý v 11 hodin ráno po dobu jedné hodiny po 3 měsíce, pak bude položka *start-time* z parametru *Timing* reprezentovat pomocí NTP hodnotu 10 hodin ráno prvního pondělí, *repeat interval* bude 1 týden, *active duration* bude 1 hodina a offset bude mít hodnotu 0 a 25 hodin. Výsledek tohoto příkladu a přesný tvar parametru je uveden zde:

Příklad: *t=3034423619 3042462419*
r=604800 3600 0 90000

⁸ MILLS, D., DELAWARE, J., MARTIN, U., BURBANK, J., KASCH, W. *Network Time Protocol Version 4: Protocol and Algorithms Specification*, RFC5905, June 2010

3.2.3.11 Time Zones (z=)

Formát: $z=<adjustment\ time> <offset> <adjustment\ time> <offset> \dots$

Volitelný parametr Time Zones slouží k tomu, abychom mohli opakovaně spouštět jistou relaci stále ve stejný čas i v případě přechodu na letní čas bez nutnosti modifikace relace. *Adjustment time* specifikuje čas, ve který je *Repeat Time* překalkulován o zadanou dobu danou částí *offset*.

Příklad: $z=2882844526\ -1h\ 2898848070\ 0.$

3.2.3.12 Encryption Keys (k=)

Formát: $k=<method>$ $k=<method>:<encryption\ key>$

Parametr Encryption Keys slouží pro přenos šifrovacích klíčů v případě přenosu přes důvěryhodný a zabezpečený kanál. Může se vyskytovat před první definicí médií, kdy je společný pro všechna média, nebo je u každého média dle potřeby. Tento parametr je v současné době definován pouze z důvodu zpětné kompatibility se staršími implementacemi a jeho použití se nedoporučuje.

Příklad: $k=prompt$ $k=base64:<encoded\ encryption\ key>$

3.2.3.13 Attributes (a=)

Formát: $a=<attribute>$ $a=<attribute>:<value>$

Tento parametr označuje atributy určené k rozšíření SDP protokolu. Ty mohou být definovány v části *Session-Level*, *Media-Level*, nebo v obou částech. Jejich počet je libovolný. Část *Media-Level* (popis média) může obsahovat řadu atributů, které jsou specifické pro dané médium a podávají o něm dodatečné informace. Atributy se mohou vyskytovat ve dvou níže uvedených formách a jejich interpretace je závislá na použitém programu. Atributy musí být zaregistrované u IANA⁹. Pokud je přijat neznámý atribut, musí být příjemcem ignorován. Seznam registrovaných atributů je uveden v 6. kapitole RFC4566¹⁰.

Příklad: $a=recvonly$ $a=author:FIT\ VUT\ v\ Brne$

⁹ Internet Assigned Numbers Authority – <http://www.iana.org>

¹⁰ HANDLEY, M., JACOBSON, V., PERKINS, C. *SDP: Session Description Protocol*, RFC4566, July 2006.

3.2.3.14 Media Description (m=)

Formát: *m=<media> <port> <proto> <fmt>*

Media Description je položkou popisující jednotlivá média tzv. elementární média streamy. Uvnitř SDP protokolu se může vyskytovat několik těchto položek, ale nemusí se v něm vyskytovat ani jedna. Tento parametr obsahuje řadu částí. První z nich označuje typ média, které je popisováno. V současné době může tato podpoložka nabývat těchto hodnot: “audio”, “video”, “application”, “text” a “message”. Druhou částí je číslo portu transportního protokolu, na který bude stream zasílán. Toto číslo souvisí s třetí položkou, jež označuje onen protokol, kterým je medium posíláno a v němž je zapouzdřeno. Nejpoužívanějšími transportními protokoly jsou UDP a RTP. Nemusí se vždy jednat o protokol transportní vrstvy. Spolu s *connection-address*, z položky *Connection Data*, a transportním protokolem nám toto číslo portu dává informace potřebné pro připojení se k danému médiu. Poslední položka *fmt* označuje formát média daného streamu. Interpretace této položky závisí na zvoleném transportním protokolu.

Příklad: *m=video 4444 udp 33*

3.2.3.15 Přehled parametrů

Zde je uveden přehled všech položek SDP. Povinné položky jsou zde zvýrazněny tučně a nepovinné hvězdičkou. Seznam obsahuje vždy písmena označující jednotlivé položky a jejich názvy. Všechny položky jsou seřazeny podle jejich povinného umístění v paketu a rozlišeny podle jejich typu.

Session-level description

v=	(protocol version)
o=	(owner/creator and session identifier)
s=	(session name)
i= *	(session information)
u= *	(URI of description)
e= *	(email address)
p= *	(phone number)
c= *	(connection information/data, not required if included in all media)
b= *	(bandwidth information)

Time description parameters (1 - N)

t=	(time the session is active)
r= *	(zero or more repeat times)
z= *	(time zone adjustments)
k= *	(encryption key)
a= *	(zero or more session attribute lines)

Media-level description (0 - N)

m= (media name and transport address)
i= * (media title)
c=* (connection information/data, optional if included at session-level)
b=* (bandwidth information)
k=* (encryption key)
a=* (zero or more media attribute lines)

3.2.3.16 Příklad

Zde uvádím příklad SDP části zachyceného informačního paketu, který byl vyslán enkodérem Teracue ENC-100. Na příkladu si kromě povinných položek můžete povšimnout šesti atributů, které se nachází v části *Session-Level description*.

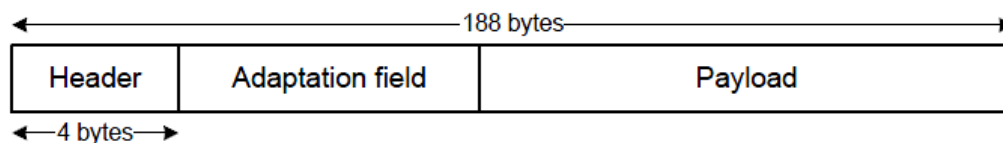
```
v=0
o=- 2208991551161452 1 IN IP4 172.16.20.100
s=Teracue
i=Live Stream
c=IN IP4 239.252.90.90/3
b=AS:5610
t=0 0
a=packetize:1316
a=packetformat:RAW
a=mux:m2t
a=keywds:Keywords
a=author:Author
a=copyright:Copyright
m=video 4444 udp 33
```

4 MPEG Transport Stream (MPEG-TS)

Tato kapitola se zaměřuje na multimediální formát MPEG Transport Stream, který se využívá pro přenos a ukládání audia a videa. Je využit také v řadě broadcastových systémů jako je DVB, ATSC a IPTV.

MPEG-TS je standard, definovaný v *ISO/IEC 13818-1¹¹*, který se používá v prostředích, kde nelze zaručit bezchybnost přenosu dat. MPEG-TS popisuje, jakým způsobem jsou části multimediálního obsahu kombinovány do jednoho datového toku pomocí tzv. *multiplexingu*. Tyto části multimediálního obsahu se nazývají *elementární toky* nebo též *elementární streamy*. Každý elementární stream obsahuje pouze jediný typ média (audio, video, text, atd.). Tyto jednotlivé streamy jsou převedeny na pakety (tzv. *PES pakety*) a multiplexovány dohromady spolu s informacemi, které zajišťují synchronizovanou prezentaci jednotlivých elementárních streamů v rámci programu.

PES pakety jsou rozděleny na pakety pevné velikosti a vkládány do *Transport Stream paketů*. Velikost těchto TS paketů je obvykle 188 Bajtů, z nichž první 4 Bajty jsou hlavička a zbylých 184 Bajtů jsou data. Tvar TS paketu je zobrazen na obrázku 3.1.



Obrázek 3.1: Formát TS paketu

Hlavička TS paketu obsahuje 13 bitový identifikátor paketu (*PID*), který identifikuje elementární stream, do kterého daná data patří. TS pakety obvykle tvoří sekvence, které jsou vždy posílány společně v UDP paketu. Uživatel, který daný UDP paket přijímá, proto musí určit, o které TS pakety má zájem a to učiní pomocí jejich *PID*. Tuto strukturu lze vidět na obrázku 3.2.

Time	Source	Destination	Protocol	Length	Info
2 0.001925	172.16.20.100	239.252.90.90	MPEG TS	1358	Source port: blackjack Destination port: krb524
3 0.003803	172.16.20.100	239.252.90.90	MPEG TS	1358	Source port: blackjack Destination port: krb524

<		>
+ Frame 3: 1358 bytes on wire (10864 bits), 1358 bytes captured (10864 bits)		
+ Ethernet II, Src: Vidisys_0b:c8 (00:50:c2:39:5b:c8), Dst: IPv4mcast_7c:5a:5a (01:00:5e:7c:5a:5a)		
+ Internet Protocol Version 4, Src: 172.16.20.100 (172.16.20.100), Dst: 239.252.90.90 (239.252.90.90)		
+ User Datagram Protocol, Src Port: blackjack (1025), Dst Port: krb524 (4444)		
+ ISO/IEC 13818-1 PID=0x103 CC=10		
+ ISO/IEC 13818-1 PID=0x103 CC=11		
+ ISO/IEC 13818-1 PID=0x100 CC=12		
+ ISO/IEC 13818-1 PID=0x100 CC=13		
+ ISO/IEC 13818-1 PID=0x100 CC=14		
+ ISO/IEC 13818-1 PID=0x100 CC=15		
+ ISO/IEC 13818-1 PID=0x103 CC=12		

Obrázek 3.2: Struktura MPEG-TS paketu

¹¹ ISO/IEC 13818-1:2007, *Information technology - Generic coding of moving pictures and associated audio information – Part 1: Systems*

Můžete si zde povšimnout, že paket zachycený programem *Wireshark*¹², v sobě obsahuje sedm MPEG-TS paketů, které jsou tvořeny z PES paketů dvou streamů s PID 0x100 a 0x103.

Pro usnadnění výběru PID jsou jako součást TS zasílána metadata s informacemi o program tzv. *Program Specific Information (PSI)*. Tyto tabulky jsou znázorněny v tabulce 3.3.

Structure Name	Stream Type	PID number	Description
Program Association Table	ITU-T Rec. H.222.0 ISO/IEC 13818-1	0x00	Associates Program Number and Program Map Table PID
Program Map Table	ITU-T Rec. H.222.0 ISO/IEC 13818-1	Assignment indicated in the PAT	Specifies PID values for components of one or more programs
Network Information Table	Private	Assignment indicated in the PAT	Physical network parameters such as FDM frequencies, Transponder Numbers, etc.
Conditional Access Table	ITU-T Rec. H.222.0 ISO/IEC 13818-1	0x01	Associates one or more (private) EMM streams each with a unique PID value
Transport Stream Description Table	ITU-T Rec. H.222.0 ISO/IEC 13818-1	0x02	Associates one or more descriptors from Table 2-39 to an entire Transport Stream

Tabulka 3.3: Program Specific Information tabulky

¹² Wireshark - <http://www.wireshark.org/>

5 Návrh programu

Tato kapitola obsahuje postup návrhu vyvíjené aplikace. Je zde popsán návrh, ke kterému jsem postupem času došel, a problémy, kterým jsem musel v průběhu návrhu čelit.

Hned na začátku bylo potřeba zvolit programovací jazyk a navrhnout strukturu programu, podle kterých byl program později napsán. Již od prvopočátku jsem měl v úmyslu vytvořit modulární konzolovou aplikaci, která by díky své modularitě mohla být jednoduše rozšiřitelná přidáním dalších modulů a jejímž hlavním výstupem by byl síťový provoz vytvořených informačních paketů. Díky mé předešlé zkušenosti s vytvářením síťových aplikací v programovacím jazyce C přišla volba právě na tento jazyk. Při vytváření struktury programu jsem navrhl několik základních modulů, jejichž funkce byly postupně rozšiřovány. Navržené moduly jsou zobrazeny v tabulce 5.1.1 a jejich podrobnějšímu popisu a vzájemnému propojení se věnuje podkapitola 5.1.

Po navržnutí modulů bylo třeba vyřešit, zda budou moduly spolupracovat sériově, či bude pro správný běh aplikace zapotřebí i jejich paralelního běhu. Podrobnější rozbor této otázky a odpověď na ni naleznete v podkapitole 5.2.

Část programu, jehož návrhu bylo věnováno velké úsilí, byla konfigurace programu. Nešlo jen o způsob nakonfigurování programu, ale záměrem bylo, aby byla konfigurační data vhodně prezentována a konfigurace programu nebyla příliš složitá. Proto jsem pro program navrhnul konfigurační soubor, jehož rozboru se týká podkapitola 5.3.

V poslední podkapitole této části 5.4 je naznačeno, jakým způsobem navržený program funguje jako celek a jakým způsobem jsou v něm využita POSIX vlákna.

Název modulu	Popis činnosti
Multicast Connect	Připojení k multicastové skupině
MPEG-TS Listener	Naslouchání a zachytávání MPEG-TS streamů
SAP Sender	Vysílání oznamovacích SAP paketů
Options	Načtení a zpracování konfigurace
HTML Output	Zobrazení výstupu programu ve formě HTML

Tabulka 5.1.1: Přehled modulů programu

5.1 Moduly

Tato podkapitola obsahuje popis a způsob propojení jednotlivých modulů, jak byly navrhнутy a později implementovány v programu. Moduly jsou zde uvedeny v pořadí, v jakém jsou programem spouštěny a jak spolu navzájem spolupracují.

5.1.1 Options

Modul Options je modulem prvním, který předchází spuštění ostatních modulů. Tento modul má na starost načtení a zpracování konfiguračního souboru, který je popsán v kapitole 5.3, a následné uložení konfiguračních dat do struktur programu, pro jejich další využití ostatními moduly.

5.1.2 Multicast Connect

Jak již název naznačuje, tento modul má na starost připojení programu do multicastových skupin. K tomu využívá multicastových adres, definovaných v konfiguračním souboru programu a multicastový protokol IGMP¹³. Program pomocí tohoto modulu pošle IGMP zprávu typu „*Membership report*“ s adresou multicastové skupiny, kterou si uloží lokální (multicastový) směrovač ve své tabulce a bude tím informován, že host v jeho síti požaduje informace a data z vybrané multicastové skupiny.

5.1.3 MPEG-TS Listener

Modul Listener má za úkol zjišťovat po připojení programu k multicastové skupině, zda se v této skupině vysílá multimediální stream protokolem MPEG-TS, o kterém by mohl program informovat. To provádí sledováním komunikace na rozhraní počítače. Využívá k tomu programový filtr, který vyfiltruje provoz, aby zachytával pouze data běžící na IP adrese a portu, které získal z konfiguračního souboru. Na této adrese by se měla nacházet pouze data multimediálního streamu. Listener dokáže počítat, kolik MPEG-TS paketů dojde na rozhraní síťové karty, a pomocí toho dokáže určit, zda daný multimediální stream běží a zda běží dle předpokladů.

¹³ROSENBERK, J., SCHULYRINE, H., CAMMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, R., SCHOOLER, E. *SIP: Session Initiation Protocol*, RFC3261, June 2002.

5.1.4 HTML Output

HTML Output slouží jako vedlejší výstup programu. Jeho hlavní činností je sbírat informace o činnosti programu a informovat o nich pomocí výstupu do souboru ve formátu HTML. Modul informuje o tom, které multimediální streamy běží po spuštění programu, jaká je jejich adresa a port pro připojení, kolik paketů daného streamu bylo zachyceno a zda je informováno o jejich běhu.

Název streamu	IP adresa	Port	Zasílat SAP pakety	Nahradit stream	Minimální počet paketů	Zachycený počet paketů	Stav streamu
D105 platno + ucitel	226.22.36.122	4448	ANO	NE	200	0	NEBĚŽÍ
A112	226.22.36.130	4464	ANO	NE	0	0	NEBĚŽÍ
E112	226.22.36.130	4444	ANO	NE	0	0	NEBĚŽÍ
G202	226.22.36.129	4464	ANO	NE	0	0	NEBĚŽÍ

Obrázek 5.1.4: Výstupní tabulka ve formátu HTML

5.1.5 SAP Sender

Tento modul slouží jako hlavní výstup programu. Modul na základě konfigurace programu informuje o běžících multimediálních streamech prostřednictvím protokolu SAP, který je popsán v kapitole 3.1. Vytvořené oznamovací SAP pakety modul vysílá na předem známou multicastovou adresu a port. Tuto adresu lze případně změnit v konfiguračním souboru programu.

5.2 Sériovost a paralelismus

Jak je vidno z podkapitoly 5.1, moduly navrhovaného programu spolu navzájem úzce souvisí. Vstup jednoho modulu často závisí na výstupu jiného, a proto je i v implementaci programu většina těchto modulů naprogramována sériově za sebou. V případě, že by program obsluhoval a informoval o jediném multimediálním streamu, mohl by za jistých okolností běžet celý plně sériově. Jelikož však program musí obsluhovat větší množství streamů a většinu modulů je zapotřebí spouštět v cyklech, bylo třeba do programu zavést paralelismus. K tomu jsem využil POSIX vláken. Způsob využití těchto vláken je popsán v kapitole 5.4.

5.3 Konfigurace

Návrhu konfigurace byla věnována poměrně dlouhá doba. Za tímto účelem byl vytvořen konfigurační soubor, z něhož jsou konfigurační data načítána. Při tvorbě tohoto souboru, jak si můžete povšimnout v jeho ukázce na obrázku 5.3.1, jsem se inspiroval mnohými konfiguračními soubory z prostředí operačního systému *GNU/Linux*, kde jsou konfigurační soubory poměrně běžné.

Navržený konfigurační soubor se dá rozdělit na dvě části. První částí je tzv. *Globální část*, v níž se nacházejí konfigurační údaje, které jsou globální pro celý program. Jsou jimi například:

- název síťového rozhraní, se kterým bude program spolupracovat,
- multicastová adresa, na kterou se budou zasílat SAP oznámení,
- doba, po jejíž uplynutí se bude opakovat sběr dat (*repeat_time*),
- umístění výstupního HTML souboru a jiné.

Druhá část konfiguračního souboru je tzv. *Programová část*. Tato část obsahuje konfigurační údaje, které jsou specifické pro každý jednotlivý stream. Nachází se zde informace potřebné pro připojení k danému streamu (IP adresa, port), informace pro popis streamu, které budou zasílány uvnitř SAP paketů (název relace, popis relace, atd.) a také informace potřebné k upřesnění běhu programu. Těmito informacemi jsou doba sběru MPEG-TS paketů modulem Listener (*listen_time*), příznak informování o streamu pomocí oznamovacích SAP paketů, příznak záměny zdroje streamu a další.

```
#####
# Konfiguracni soubor programu atmas2013          #
# Autor: Petr Dorusak - XDORUS01                 #
#####
[GLOBAL]
interface=eth1
listen_time=10
repeat_time=10
send_address=224.2.127.254
output_path=/home/petr/Plocha/index.html
#####
[PROGRAM]
name=A112
user=FIT
description=Stream z prednasky z ucebny VUT
machine=agnes.fit.vutbr.cz
address=226.22.36.130
port=4464
listen_time=6
sap_enabled=1
sap_delay=3
switch_mode=1
category=TEST
switch_source_addr=224.22.22.22
switch_source_port=1111
#####
```

Obrázek 5.3.1: Ukázka konfiguračního souboru

5.4 Princip programu

Program, jehož princip je v této kapitole naznačen, je celek tvořený pomocí jednotlivých modulů a POSIX vláken, jež s těmito moduly pracují. Je zde tedy uvedeno, jakým způsobem jsou v programu POSIX vlákna využita, ve kterém časovém okamžiku vznikají a jakým způsobem pracují s jednotlivými moduly. Dále je zde zmínka o vybraném synchronizačním prostředku pro správnou spolupráci vláken.

5.4.1 Start programu

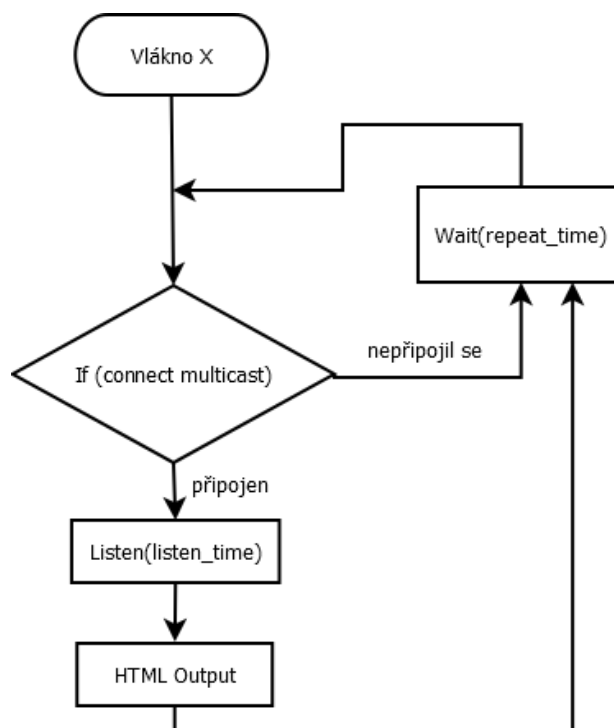
Startovací sekvence programu vždy začíná stejně. Program aktivuje modul Options k načtení a zpracování konfiguračního souboru, jehož položky si uloží do svých vnitřních struktur, aby s nimi mohl dále pracovat. Po této startovací fázi se program dostane k vytvoření POSIX vláken.

5.4.2 Vlákna a jejich synchronizace

Po načtení konfigurace do struktur programu se začne s tvorbou vláken. Program má zadán počet streamů, o které se bude muset starat a informovat o nich. Podle tohoto počtu se vytvoří první sada vláken. Tato vlákna budou následně kontrolovat, zda streamy běží a budou informovat o jejich stavu. Sekvence jejich prováděných úkonů je znázorněna diagramem a obrázkem 5.4.2.

Paralelně s těmito vlákny se po jejich prvním průchodu smyčkou vytvoří stejný nebo nižší počet vláken, která se starají o zasílání oznamovacích paketů. Počet těchto vláken závisí na konfiguraci streamů. Vlákna se vytvoří jen pro ty streamy, pro které je povoleno oznamování pomocí protokolu SAP.

Každé vlákno dle aktuálního stavu a konfiguračního nastavení periodicky oznamuje běh jednoho z těchto streamů s periodou nastavenou konfiguračním souborem. V programu je tedy generováno maximálně $2*n$ vláken, kde n je počet všech streamů nacházejících se v konfiguračním souboru.

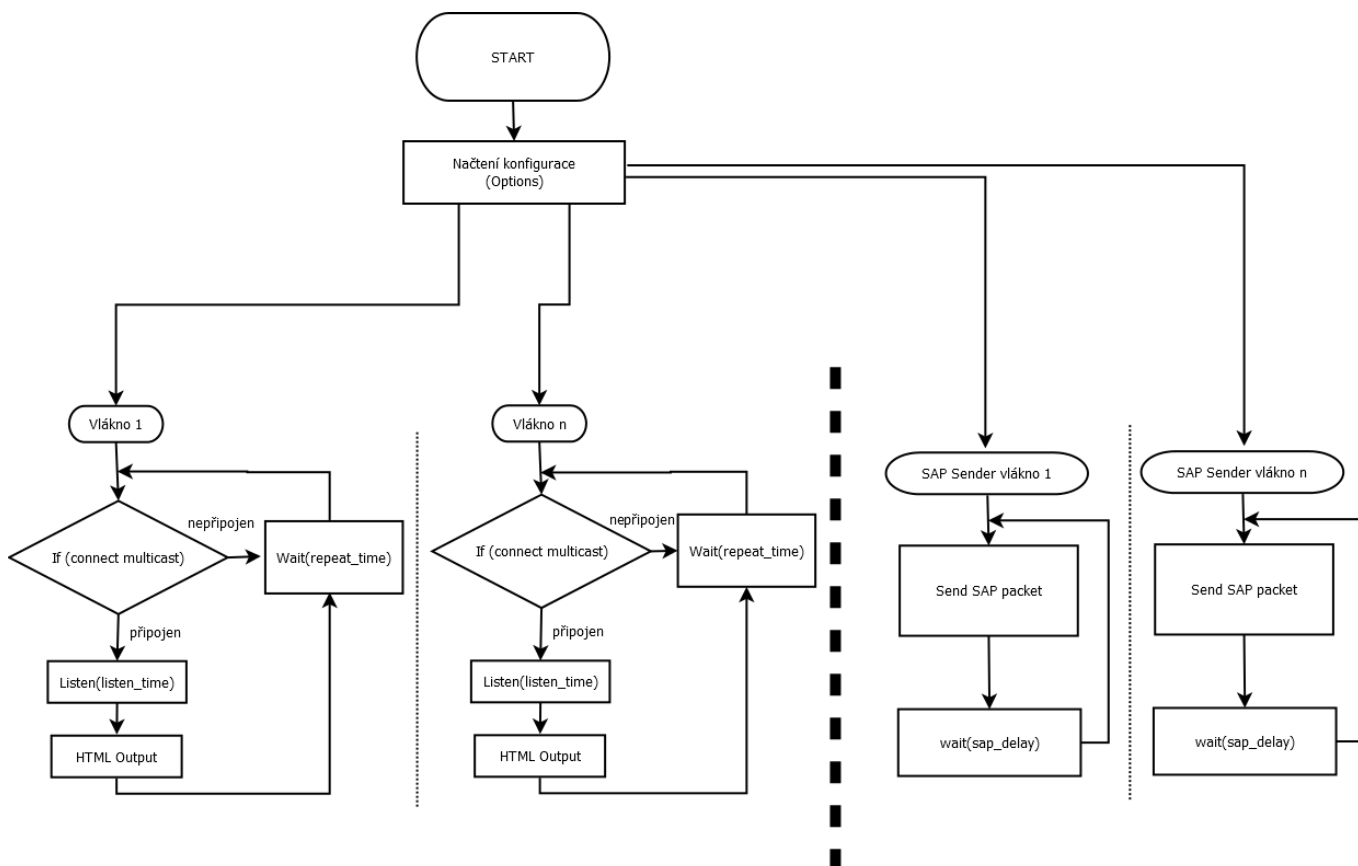


Obrázek 5.4.2: Diagram úkonů stream vláken

Jelikož bylo při návrhu rozhodnuto, že budou v programu využita POSIX vlákna, jež některá budou zapisovat a jiná číst stejná data, bylo zapotřebí určit vhodný synchronizační mechanismus, aby spolu vlákna vhodně spolupracovala. Pro tento účel jsem se rozhodl pracovat s jedním z nejjednodušších mechanismů - s *mutex zámkami*. Díky nim jsem zajistil, že se k určité položce dostalo v jednom časovém okamžiku pouze jediné vlákno, které mohlo s položkou pracovat bez rizika kolize dat.

5.4.3 Celková podoba programu

Jak už bylo zmíněno v předchozí podkapitole, v programu běží dvě skupiny vláken, z nichž jedna se stará o sběr informací a druhá o výstup programu. Program je navržen jako nikdy nekončící smyčka, která může být ukončena pouze vnějším zásahem uživatele. Vzhledem k této podobě a faktu, že se jedná o konzolovou aplikaci, jejímž výstupem je síťový provoz a soubor ve formátu HTML, je vhodné spustit tento program na pozadí operačního systému. Aplikace bude bez zásahu uživatele vykonávat svou činnost a nebude jej rušit v práci. Celkový běh programu je znázorněn na diagramu 5.4.3.



Obrázek 5.4.3 Diagram běhu programu

6 Implementace programu

Jak jsem již zmínil v kapitole 5, aplikace je implementována v programovacím jazyce C, je spustitelná na operačním systému GNU/Linux a pro správný běh vyžaduje oprávnění superuživatele *root*. V této kapitole popisují několik zajímavých konstrukcí, které jsem použil při její implementaci a které jsou vhodné pro lepší pochopení její funkce.

6.1 Zachytávání paketů

Pro zachytávání MPEG-TS paketů je v aplikaci využita síťová knihovna *libpcap*. Pomocí této knihovny je v programu otevřeno síťové rozhraní pro zachytávání paketů a je vytvořen a zkompilován paketový filtr pro jejich filtrování. Před zkompilováním má filtr podobu textového pole, a proto je pro jeho nastavení třeba zadat jen správná klíčová slova. Podoba mého filtru je naznačena v pseudokódu 6.1.1. Filtr propouští pouze UDP pakety nacházející se na multicastové adrese (*stream_address*) a portu (*stream_port*) každého ze streamů, které jsou zadány v konfiguračním souboru.

```
filtr="udp and dst net stream_address and dst port stream_port";
```

Obrázek 6.1.1 – Pseudokód paketového filtru

Po otevření rozhraní a zkompilování filtru je filtr aplikován a začne sběr paketů. Ten je prováděn ve while cyklu a je ukončen při překročení času sběru paketů.

```
pcap_t *desc; //popisovač relace
/*Otevřu zařízení pro naslouchání*/
descr=pcap_open_live(interface,BUFSIZ,1,ptr_stream->listen_time,errbuf);
/*Zjistím si masku a ip adresu zařízení*/
pcap_lookupnet(interface, &ipa, &maska, errbuf);
/*Zkompiluji filtr*/
pcap_compile(descr, &fp, filtr, 0, maska);
/*Aplikuji filtr*/
pcap_setfilter(descr, &fp);
/*Smyčka sběru*/
while((pcap_packet == 1) &&
      (current_time < start_time + ptr_stream->listen_time)){
    /*Seberu paket*/
    pcap_packet = pcap_next_ex(descr,&header,(const u_char **) &data);
    /*Zavolám callback funkci pro jeho zpracování*/
    packet_count= pcap_dispatch(descr, -1, got_packet, NULL);}
```

Obrázek 6.1.2 – Princip zachytávání paketů

V případě, že na rozhraní nebyl zachycen žádný paket, je multimediální stream označen za neběžící. V opačném případě je zkontrolováno, zda byl zachycen vyšší počet paketů, než je uveden v položce *minimum_packet_count* v konfiguračním souboru. Pokud tomu tak je, nebo pokud položka zadána nebyla, je stream označen jako běžící.

6.2 Generování a zasílání SAP paketů

Pro generování SAP paketů je v programu vytvořena funkce z názvem *create_sap_packet*, jejímž úkolem je naplnit strukturu *sap_packet* korektními informačními daty. Funkce převezme data obsažená v konfiguračním souboru a doplní je o další informace, aby vytvořila validní informační pakety, které budou následně posílány.

```
typedef struct sap_packet{
    sap_prt sap;          /* SAP part          */
    char* sdp;            /* SDP part          */
}sap_packet;

typedef struct sap_prt{
    int vartec;           /* VARTEC            */
    int auth_len;         /* Auth Length       */
    u_short hash;         /* Packet Hash       */
    char* orig_src;       /* Originating Source */
    char* payload_type;   /* Payload Type      */
}sap_prt;
```

Obrázek 6.2.1 – Struktura *sap_packet*

Pro zasílání takto naplněné struktury po síti je v programu vytvořena funkce *send_sap* využívající BSD sockety, pomocí nichž je přenos po síti uskutečněn. Data jsou nejprve vložena do předem vytvořeného bufferu, poté je vytvořen a nastaven socket a následně jsou data v cyklu zasílána po síti. Četnost zasílání paketů je dána položkou *sap_delay* konfiguračního souboru. Princip zasílání paketů je naznačen v pseudokódu na obrázku 6.2.2.

```

int send_sap(struct multicast_stream* ptr_stream, char* send_address,
            int ptr_i){
    buffer=(char*) malloc (sizeof(ptr_stream->packet) * sizeof(char));
    /*vlození dat do bufferu, příklad*/
    buffer[0]=ptr_stream->packet->sap.vartec;
    buffer[1]=ptr_stream->packet->sap.auth_len;
    /*...*/
    /* vytvoreni socketu */
    socketDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    /*nastaveni rodiny protokolu a cilove adresy a portu*/
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = inet_addr(send_address);
    serverAddress.sin_port = htons((send_port));
    while(1)
    {
        /* zaslani paketu*/
        sendto(socketDescriptor, buffer, posun, 0, (struct sockaddr*)
            &serverAddress, sizeof(serverAddress ))
        sleep(ptr_stream->sap_delay); /*cekani na dalsi zaslani*/
    }
}

```

Obrázek 6.2.2 – Pseudokod funkce send_sap

6.3 Nahrazování zdroje streamů

Implementovaný program umožňuje kromě zasílání standardních informačních paketů i zasílání paketů se změněným zdrojem vysílání. Tuto možnost lze aktivovat v konfiguračním souboru u každého streamu pomocí příznaku *switch_mode*.

```

#####
[PROGRAM]
name=D105 platna + ucitel
user=FIT
description=Stream z prednasky z ucebny VUT
address=226.22.36.123
port=4450
sap_enabled=1
sap_delay=5
switch_mode=0  Příznak změny zdroje
playlist_group=FIT_STREAMS
#####

```

Obrázek 6.3.1 – Ukázka příznaku nahrazení streamu

Je-li tento příznak nastaven a oznamovaný stream přestane vysílat, program modifikuje zasílané informační pakety. Začne zasílat pakety s náhradním zdrojem streamu zadaným adresou a portem v položkách *switch_source_addr* a *switch_source_port* konfiguračního souboru. Vysílání modifikovaných paketů je prováděno do doby, než je originální stream opět zapnut. Poté jsou pakety upraveny zpět na originální zdroj.

Výše uvedenou adresu a port náhradního zdroje lze v konfiguračním souboru definovat jednou společně pro všechny streamy v globální sekci, nebo samostatně pro každý jednotlivý stream v programových sekcích. Díky tomuto způsobu konfigurace lze nahrazování zdroje využít několika způsoby.

První možností je redundance. V případě, že by program informoval o nějaké důležité události vysílané prostřednictvím několika multicastových streamů (projev prezidenta republiky, sportovní utkání), by bylo vhodné zvolit jeden ze streamů jako primární, o kterém by program informoval, a dále zvolit některý z dalších streamů jako sekundární. Adresa a port sekundárního streamu by byly vedeny v položkách *switch_source_addr* a *switch_source_port*. V případě výpadku primárního streamu by program automaticky informoval o téže události prostřednictvím sekundárního zdroje do doby, než by došlo k obnovení primárního zdroje.

Další možností je využití pro informační či reklamní účely. Pokud bude program informovat o časově omezených opakujících se streamech (televizní vysílání, stream přednášek v rámci počítačové sítě VUT), je možné po ukončení jejich vysílání nahradit zdroj jiným reklamním či informačním obsahem, který je vysílán prostřednictvím multicastového streamu, jež bude nahrazovat právě neběžící stream.

Celý systém nahrazování je v programu vyřešen za pomoci sady cyklů a podmínek, v nichž se kontroluje, zda stream běží a zda se nemají vysílané pakety modifikovat. Nahrazování streamu je potom patrné z výstupní tabulky zobrazené na obrázku 6.3.2. Zde uvedený stream *D105 platno+ucitel*, dle posledního sloupce tabulky neběží a v příznaku Nahradit stream, který reflektuje proměnnou konfiguračního souboru *switch_mode*, má nastavenou hodnotu ANO. V okamžiku zobrazení této tabulky je tedy oznamován stream s náhradním zdrojem.

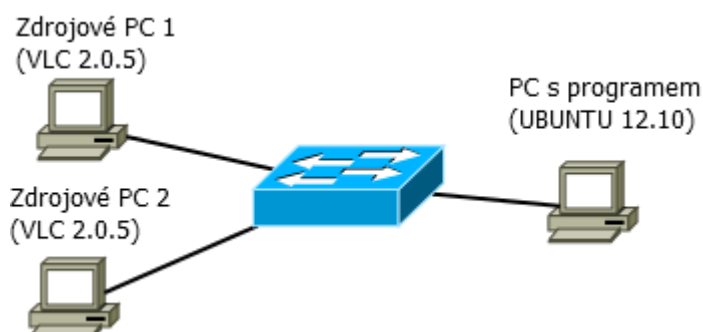
Název streamu	IP adresa	Port	Zasílat SAP pakety	Nahradit stream	Minimální počet paketů	Zachycený počet paketů	Stav streamu
D105 platno + ucitel	226.22.36.122	4448	ANO	ANO	5	0	NEBĚŽÍ
A112	226.22.36.130	4464	ANO	ANO	0	0	NEBĚŽÍ
E112	226.22.36.130	4444	ANO	ANO	0	0	NEBĚŽÍ
G202	226.22.36.129	4464	NE	ANO	0	0	NEBĚŽÍ

Obrázek 6.3.2 – Ukázka nahrazeného streamu

7 Testování

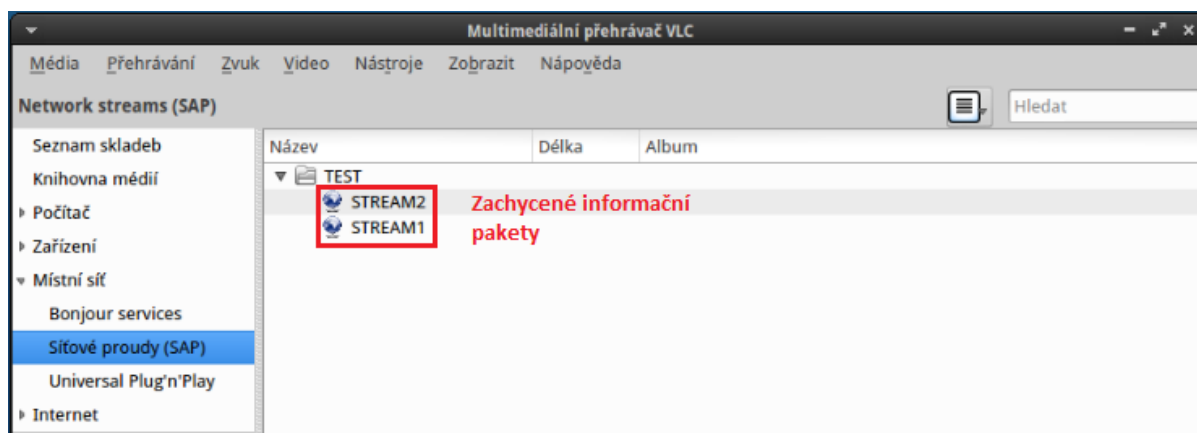
7.1 Testování modulů

Pro ověřování funkčnosti nástroje probíhalo v průběhu jeho vývoje opakované testování. Prvotní testování probíhalo na předem vytvořené lokální síti obsahující dva počítače s operačním systémem Windows XP, které pracovaly jako zdroje streamu, a jedním počítačem s operačním systémem Ubuntu 12.10, na kterém běžela vlastní aplikace (obrázek 7.1.1). Na počítačích s Windows XP běžel multimediální přehrávač VLC¹⁴, s jehož pomocí byly vytvořeny multicastové multimediální streamy, které byly následně zasílány po síti.



Obrázek 7.1.1 – Testovací síť

Úkolem aplikace bylo nejprve tyto streamy zachytit a následně informovat o jejich běhu vysíláním paketů protokolu SAP. Zda program fungoval a pakety byly vysílány, bylo zkontrolováno programem Wireshark a další instancí programu VLC, který umí zachytávat informační pakety protokolu SAP (obrázek 7.1.2). V případě, že stream nebyl zachycen, se jednalo o chybu určitého modulu a bylo potřeba zjistit chybu pomocí výstupního logu aplikace.

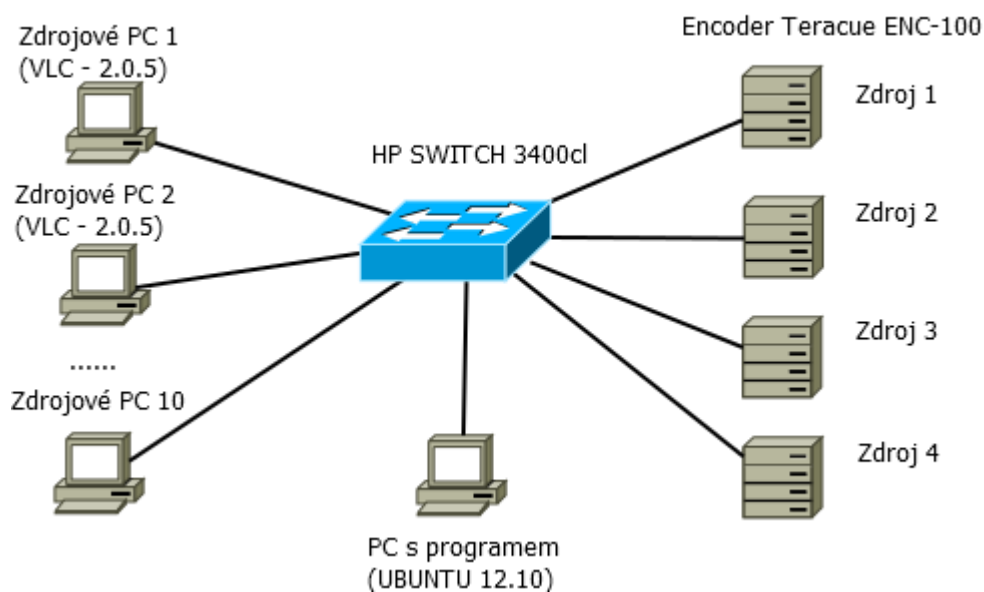


Obrázek 7.1.2 – Zachycení SAP paketů programem VLC

¹⁴ VLC media player - <http://www.videolan.org/vlc/>

7.2 Zátěžové testování

Po dokončení aplikace a po otestování všech modulů bylo provedeno zátěžové testování. Úkolem bylo zjistit, zda aplikace dokáže pracovat s větším množstvím streamů, které jsou vysílány s vyšším datovým tokem. Pro tento test byla vytvořena síť v počítačové učebně L307 na fakultě Informačních technologií VUT v Brně, která obsahovala deset počítačů s operačním systémem Windows XP a čtyři enkodéry Teracue ENC-100, které sloužily jako zdroje streamu (obrázek 7.2.1).



Obrázek 7.2.1 – Testovací síť zátěžového testu

Na všech počítačích byly opět za pomoci programu VLC vytvořeny multicastové streamy, které byly zasílány po síti a které byly podpořeny dalšími čtyřmi streamy z výše jmenovaných enkodérů. Celkový datový tok při zapnutí všech streamů činil přibližně 26 Mbit/s. Přehled jednotlivých datových toků je znázorněn tabulkou 7.2.2.

Zařízení	Datový tok	Celkový datový tok
PC	1 Mbit/s	10 x 1 = 10 Mbit/s
Enkodér 1	1,5 Mbit/s	1,5 Mbit/s
Enkodér 2	5,61 Mbit/s	5,61 Mbit/s
Enkodér 3	5,61 Mbit/s	5,61 Mbit/s
Enkodér 4	3,54 Mbit/s	3,54 Mbit/s
Celkem		26,26 Mbit/s

Při testu byly testovány reakce programu na zapínání a vypínání jednotlivých streamů a schopnost vypořádat se s větším datovým tokem. Na závěr testování byl program spuštěn se zachytáváním paketů nastaveným na dobu pěti minut. Při takto velkém datovém toku se dalo očekávat, že program zachytí velké množství paketů a hrozil jeho pád z důvodu přetížení. Program však tímto testem prošel a důkazem jeho činnosti je výstupní tabulka na obrázku 7.2.3, z níž vyplývá, že program z devíti běžících streamů zachytil celkově přes 428 tisíc paketů.

Tabulka informací o nastavených streamech

Název streamu	IP adresa	Port	Zasílat SAP pakety	Nahradit stream	Minimalní počet paketů	Zachycený počet paketů	Stav streamu
STREAM1	226.0.0.1	4001	ANO	NE	50	0	NEBĚŽÍ
STREAM2	226.0.0.2	4002	ANO	NE	50	0	NEBĚŽÍ
STREAM3	226.0.0.3	4003	ANO	NE	300	30089	BĚŽÍ
STREAM4	226.0.0.4	4004	ANO	NE	50	0	NEBĚŽÍ
STREAM5	226.0.0.5	4005	ANO	NE	50	33149	BĚŽÍ
STREAM6	226.0.0.6	4006	ANO	NE	50	34048	BĚŽÍ
STREAM7	226.0.0.7	4007	ANO	NE	0	33825	BĚŽÍ
STREAM8	226.0.0.8	4008	ANO	NE	0	34114	BĚŽÍ
STREAM9	226.0.0.9	4009	ANO	NE	0	33806	BĚŽÍ
STREAM10	226.0.0.10	4010	ANO	NE	0	33728	BĚŽÍ
STREAM11	226.0.0.11	4011	ANO	NE	0	41370	BĚŽÍ
STREAM12	226.0.0.12	4012	ANO	NE	0	0	NEBĚŽÍ
STREAM13	226.0.0.13	4013	ANO	NE	0	154680	BĚŽÍ
STREAM14	226.0.0.14	4014	ANO	NE	0	0	NEBĚŽÍ

Obrázek 7.2.3 – Výstupní tabulka programu

8 Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat nástroj, který bude sledovat MPEG-TS provoz v zadaných multicastových skupinách a bude informovat o existenci multicastových streamů pomocí protokolu SAP. Byla navrhována a implementována modulární aplikace reflektující zmíněné požadavky, kterou je možné spustit na pozadí operačního systému.

Aplikace byla testována v laboratorním prostředí na operačním systému GNU/Linux, přičemž byly nalezeny a opraveny drobné chyby. Při výsledném otestování tedy probíhala její činnost dle očekávání a neobjevily se žádné nové problémy. Je však škoda, že se aplikaci nepodařilo otestovat na reálném systému fakultní sítě. Testování aplikace totiž probíhalo mimo dobu streamování přednášek a bylo by nutné, tak jako v laboratoři, generovat provoz na fakultní síti vlastnoručně.

Po důkladném otestování na reálném systému by bylo možné využít aplikaci v prostředí fakultní sítě jako náhradu za stávající informování o streamech prostřednictvím enkodérů z přednáškových místností, což by umožnilo centrální správu informací o jednotlivých přednáškách a kontrolu běhu streamů prostřednictvím přehledné tabulky generované ve formátu HTML.

Pro další rozvoj aplikace by bylo vhodné se zaměřit zejména na to, aby bylo možné aplikaci použít v IPv6 sítích, pro jejichž architekturu není program nyní stavěn a dále zvážit možnost vytváření vlastního multimediálního streamu, který by program vysílal jako náhradní zdroj.

Literatura

- [1] CAIN, B., DEERING, S., FENNER, B., KOUVELAS, I. *Internet Group Management Protocol, Version 3*, RFC3376, October 2002.
- [2] HANDLEY, M., JACOBSON, V., PERKINS, C. *SDP: Session Description Protocol*, RFC4566, July 2006.
- [3] HANDLEY, M., PERKINS, C., WHELAN, E. *Session Announcement Protocol*, RFC2974, October 2000.
- [4] HARTE, Lawrence. *Introduction to data multicasting*. Fuquay-Varina: Althos Publishing, c2008, viii, 72 s. ISBN 19-328-1355-1.
- [5] ISO/IEC 13818-1:2007, *Information technology - Generic coding of moving pictures and associated audio information – Part 1: Systems*.
- [6] KOSIUR, David. *IP multicasting*. Vyd. 1. New York: J. Wiley, 357 s. ISBN 0-471-24359-0.
- [7] MAYER, D. *Administratively scoped IP multicast*, RFC2365, July 1998.
- [8] ROSENBERK, J., SCHULYRINE, H., CAMMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, R., SCHOOLER, E. *SIP: Session Initiation Protocol*, RFC3261, June 2002.

Seznam příloh

Příloha A. Návod k použití

Příloha B. CD/DVD

Příloha A

Návod k použití

Překlad aplikace

K překladu zdrojových kódů aplikace na operačním systému GNU/Linux slouží přiložený *Makefile* soubor, který pro správné přeložení aplikace vyžaduje dostupnou knihovnu *libpcap*. Vlastní překlad se provede příkazem *make*, čímž se jednotlivé soubory přeloží a spojí se do spustitelného souboru.

Na jiných operačních systémech nemusí být přiložený *Makefile* kompatibilní, a proto se doporučuje přeložit program ručně pomocí následující sekvence příkazů:

```
gcc -c main.c -std=gnu99 -Wall -g3
gcc -c options.c -std=gnu99 -Wall -g3
gcc -c listener.c -std=gnu99 -Wall -g3
gcc -c sender.c -std=gnu99 -Wall -g3
gcc -c multicast-connect.c -std=gnu99 -Wall -g3
gcc -c html.c -std=gnu99 -Wall -g3
gcc main.o listener.o options.o multicast-connect.o sender.o
html.o -lpcap -lpthread -o atmas2013
```

Konfigurace

Konfigurace programu je uskutečněna prostřednictvím konfiguračního souboru. Program obsahuje ve složce se zdrojovými soubory výchozí konfigurační soubor s názvem *options.conf*, se kterým pracuje. Tento soubor lze nahradit vlastním konfiguračním souborem spuštěním aplikace s parametrem *-c*. Použití tohoto parametru je naznačeno v následující kapitole.

Konfigurační soubor se skládá ze dvou částí: Globální a Programové. Globální část obsahuje konfigurační údaje, které jsou globální pro celý program. Začíná klíčovým slovem *[GLOBAL]* a končí první programovou sekcí. Programová část obsahuje údaje, které jsou specifické pro každý stream. Začíná klíčovým slovem *[PROGRAM]* a končí jinou programovou sekcí nebo koncovou zářátkou konfiguračního souboru *[END]*. V následujících podkapitolách budou popsány všechny povinné i volitelné položky konfiguračního souboru. Všechny položky jsou ve tvaru TYP=HODNOTA, jsou citlivé na velikost písmen, a proto je nutné zadávat název položky malými písmeny.

Globální část

Povinné:

- *interface* název rozhraní síťové karty, se kterým bude program spolupracovat a na které bude program zasílat vytvořené informační pakety.
- *listen_time* doba sběru MPEG-TS paketů modulem listener uvedená v sekundách. Tato hodnota je použita v případě nespecifikování doby sběru v programové části.
- *repeat_time* doba mezi jednotlivými sběry, po kterou program počká, než začne znovu sbírat MPEG-TS pakety. Hodnota je uvedena v sekundách.
- *send_address* multicastová adresa, na niž budou zasílána oznámení ve formě SAP paketů.

Volitelné:

- *output_path* cesta pro umístění výstupního HTML souboru. Pokud není parametr zadán, je soubor uložen do podadresáře */html* v umístění binárního souboru.

```
#####  
# Konfiguracni soubor programu atmas2013 #  
# Autor: Petr Dorusak - XDORUS01 #  
#####  
[GLOBAL]  
#navez rozhrani sitove karty (REQUIRED)  
interface=eth1  
  
#Doba sberu paketu (s) (listen_time > 0) (REQUIRED)  
listen_time=10  
  
#Cas, po kterem se bude sber paketu opakovat (REQUIRED)  
# (repeat_time > 0)  
repeat_time=10  
  
#Multicastova adresa, SAP oznameni (REQUIRED)  
send_address=224.2.127.254  
  
#Umisteni vystupniho html souboru (OPTIONAL)  
#Defaultne nastaven na podadresar /html v projektu  
#output_path=/home/petr/Flocha/index.html  
#####
```

Obrázek 8.1.2.1 – Globální část konfiguračního souboru

Programová část

U vybraných položek je v závorce uvedeno písmeno parametru protokolu SDP, kterou tyto položky reprezentují.

Povinné:

- *name* název streamu. (*s=*)
- *machine* DNS záznam nebo IP adresa zdroje streamu. (*o=*)
- *address* IP adresa multicastové skupiny, na níž se nachází zkoumaný stream. (*c=*)
- *port* port multicastové skupiny, na němž se nachází zkoumaný stream. (*m=*)
- *user* jméno vlastníka zkoumaného multicastového streamu. (*o=*)
- *sap_delay* interval zasílání informačních paketů v sekundách.
- *sap_enabled* příznak vytvoření a zasílání informačních paketů. Má-li příznak nulovou hodnotu, program pouze kontroluje, zda zadaný stream běží, ale neinformuje o jeho běhu.
- *switch_mode* příznak náhradního zdroje streamu. Pokud je příznak aktivní (*=1*) a kontrolovaný stream přestane běžet, program začne informovat o náhradním zdroji streamu zadaným adresou *switch_source_addr* a portem *switch_source_port*.

Volitelné:

- *description* textový popis streamu. (*i=*)
- *uri* adresa sloužící k podání dalších informací o streamované relaci. (*u=*)
- *email* e-mailová adresa na osobu zodpovědnou za vysílání stream. (*e=*)
- *bandwidth* předpokládaný datový tok streamu. (*b=*)
- *charset* kódování textových informací o streamu. (*a=charset*)
- *keyword* klíčová slova popisující stream. (*a=keywds*)
- *copyright* jméno autora či držitele autorských práv. (*a=author, a=copyright*)
- *category* kategorie určená pro třídění oznámení o streamech. (*a=cat*)
- *sap_ttl* nastavení hodnoty TTL oznamovacích paketů. Výchozí hodnota je 255. (*c=*)
- *listen_time* doba sběru MPEG-TS paketů specifická pro daný stream.
- *repeat_time* doba mezi sběry MPEG-TS paketů specifická pro daný stream.
- *minimum_packet_count* minimální počet paketů, které musí program zachytit během jednoho sběru paketů, aby označil stream za běžící.
- *switch_source_addr* adresa náhradního zdroje streamu. Výchozí hodnota je 127.0.0.1.
- *switch_source_port* port náhradního zdroje streamu. Výchozí hodnota je 5000.

```
#####
[PROGRAM]
name=A112
user=FIT
description=Stream z prednasky z ucebny VUT
machine=agnes.fit.vutbr.cz
address=226.22.36.130
port=4464
sap_enabled=1
sap_delay=3
switch_mode=1
category=TEST
switch_source_addr=224.22.22.22
switch_source_port=1111
#####
```

Obrázek 8.1.2.2 – Programová část konfiguračního souboru

Spuštění aplikace

Přeložený binární soubor *atmas2013* lze spustit několika způsoby a je nutné ho spustit s oprávněním superuživatele *root*. Prvním způsobem je spuštění jako běžné aplikace pomocí příkazu *./atmas2013*. Výstupní informace o běhu programu jsou pak vypisovány na standardní výstup a chybová hlášení na standardní chybový výstup. Program pracuje defaultně v tzv. Tichém módu, kdy nevypisuje žádné informace o svém běhu. Pro výpis kontrolních informací je nutno program spustit s parametrem *-p*.

Parametry programu:

```
atmas2013 [-p] [-h] [-c PATH]
```

- p výpis kontrolních informací o běhu programu.
- c nastavení konfiguračního souboru programu, při nezadání je použit soubor *options.conf* umístěný ve složce programu.
- h nápověda programu.

Druhým způsobem je spuštění za pomoci vytvořeného spouštěcího skriptu s názvem *atmas.sh*. Tento skript automaticky spustí program na pozadí operačního systému v režimu výpisu kontrolních informací a přesměruje jeho standardní výstup do logovacího souboru *out.log* a standardní chybový výstup do souboru *err.log*. Skript lze dále použít k vypnutí či restartování aplikace.

Parametry skriptu:

```
atmas.sh {start|stop|force-stop|restart} [-c PATH]
```

start	spuštění aplikace.
stop	vypnutí aplikace.
force-stop	násilné vypnutí aplikace.
restart	restartování aplikace.
-c	nastavení konfiguračního souboru programu .

Příklad spuštění programu:

<code>sudo ./atmas2013 -c test.conf</code>	Spuštění programu s konfiguračním souborem s názvem <i>test.conf</i> .
<code>sudo ./atmas2013 -p</code>	Spuštění programu v režimu výpisu kontrolních informací na standardní výstup.
<code>sudo ./atmas.sh start</code>	Spuštění programu na pozadí OS.
<code>sudo ./atmas.sh start -c test.conf</code>	Spuštění programu na pozadí OS s konfiguračním souborem <i>test.conf</i> .

Návratová hodnota aplikace

V případě ukončení aplikace ať už z důvodu ukončení uživatelem či v případě chyby, program vrací návratovou hodnotu. Dle návratové hodnoty lze vyčíst, zda byl program ukončen bez chyby, či o jakou chybu programu se jednalo. Přehled návratových hodnot a jejich popis je v následující tabulce:

Návratová hodnota	Význam návratové hodnoty
0	Bez chyby
1	Neznámý parametr příkazového řádku
2	Chyba modulu Sender
3	Chyba modulu Listener
4	Chyba modulu Multicast Connect
5	Chyba modulu HTML Output
6	Chyba při zpracování konfiguračního souboru
7	Chyba hlavní funkce programu

Tabulka návratových hodnot programu

Příloha B

Obsah CD

- Bakalářská práce v elektronické podobě
- Zdrojové kódy aplikace
- Binární soubory aplikace